Assignment week 44: Webscraping

Christoffer M. Kramer

30/10/2020

My repo: <https://github.com/Digital-Methods-HASS/au590388_Christoffer_Kramer/tree/master/learning_journal_and_assignments/week_44_webscrape>

**Take the repository at** [**https://github.com/adivea/KilledbyPolice2020.git**](https://github.com/adivea/KilledbyPolice2020.git) **and depending on your familiarity with R, either**

1. *adapt the webscraping example to scrape homicide data from FBI site and produce a meaningful report on how homicide trends evolve around US in relation to this urban unrest*
2. *use the rvest library to scrape data of your interest (football statistics in Wikipedia? global population by country in* [*https://www.worldometers.info/world-population/population-by-country/*](https://www.worldometers.info/world-population/population-by-country/) *)*
3. *produce data visualisations that shed light on another interesting aspect of the police killing data*

I will do the second item on the list. I will scrape transcripts of presidential debates, since these will be usefull for my final project.

I start by loading the relevant libraries

library(rvest)

## Loading required package: xml2

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)

## Warning: package 'tidyr' was built under R version 4.0.3

library(stringr)  
library(janitor)

## Warning: package 'janitor' was built under R version 4.0.3

##   
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.3

## -- Attaching packages ------------------------------------------------------------------------------------------------ tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.3 v forcats 0.5.0  
## v readr 1.3.1

## Warning: package 'ggplot2' was built under R version 4.0.3

## -- Conflicts --------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x readr::guess\_encoding() masks rvest::guess\_encoding()  
## x dplyr::lag() masks stats::lag()  
## x purrr::pluck() masks rvest::pluck()

library(tidytext)

## Warning: package 'tidytext' was built under R version 4.0.3

library(ggplot2)

Then I create af function for webscraping. This makes it possible to automate the webscraping process. The parameter is the website I wish to scrape. The function reads the html elements of the website. It only reads the “p” elements, since this is where the text is located. It then parses the elements as text. Lastly, it saves the output in a vector, which is usefull for text-mining and data-cleaning.

scrape\_debates <- function(website) { #Create a function called scrape\_debates  
 p\_html <- read\_html(website) %>% #Create an object which contains the parameter "website" and is read as a html file  
 html\_nodes("p") %>% #Grab all <p> elements in the html file.  
 html\_text() #Parse the result as text.  
   
vect\_p\_html <- c(p\_html) #Save the output in a vector.  
}

Let’s test it:

head(scrape\_debates("https://www.debates.org/voter-education/debate-transcripts/2008-debate-transcript/"))

## [1] "September 26, 2008"   
## [2] "The First McCain-Obama Presidential Debate"   
## [3] "FIRST PRESIDENTIAL CANDIDATES’ DEBATETHE UNIVERSITY OF MISSISSIPPI, OXFORD MISSISSIPPI"  
## [4] "SPEAKERS:"   
## [5] "U.S. SENATOR JOHN MCCAIN (AZ)REPUBLICAN PRESIDENTIAL NOMINEE"   
## [6] "U. S. SENATOR BARACK OBAMA (IL)DEMOCRATIC PRESIDENTIAL NOMINEE"

tail(scrape\_debates("https://www.debates.org/voter-education/debate-transcripts/2008-debate-transcript/"))

## [1] "For now, from Oxford, Mississippi, thank you, senators, both. I’m Jim Lehrer. Thank you, and good night."  
## [2] "(APPLAUSE)"   
## [3] "END"   
## [4] "Transcription by: CQ Transcriptions/Morningside"   
## [5] "Â "   
## [6] "© COPYRIGHT 2020 THE COMMISSION ON PRESIDENTIAL DEBATES. ALL RIGHTS RESERVED."

It works, but I still have 40 transcripts that I need to scrape. I will, therefore, use a loop to automate the process.

The links to all transcripts are located here: <https://www.debates.org/voter-education/debate-transcripts/> Therefore, I need to grab all the relevant links (without getting links from the nav-bar).

Luckily the links follow an easy structure. Every link is contained within an “a” element, which is inside of of “p” element, which inside a “blockqoute” element. So by navigating to the “blockqoute” element, I can grab the “child” (which is “p”) and then navigate to the relevant “a” element and then select all a elements that contain a link (a href attribute).

link\_html <- read\_html("https://www.debates.org/voter-education/debate-transcripts/") %>%   
 html\_nodes("blockquote") %>% #Grab the blockqoute elements  
 html\_children() %>% #Navigate to the blockqoute's children.  
 html\_node("a") %>% #Grab <a> elements  
 html\_attr("href") #Grab <a> elements which contains a hyperlink.  
  
vect\_link <- c(link\_html) #save the output in a vector

Let’s see if it works.

vect\_link

## [1] "https://www.debates.org/voter-education/debate-transcripts/september-29-2020-debate-transcript/"   
## [2] "http://debates.org/voter-education/debate-transcripts/vice-presidential-debate-at-the-university-of-utah-in-salt-lake-city-utah/"  
## [3] "https://www.debates.org/voter-education/debate-transcripts/october-22-2020-debate-transcript/"   
## [4] "/voter-education/debate-transcripts/september-26-2016-debate-transcript/"   
## [5] "/voter-education/debate-transcripts/october-4-2016-debate-transcript/"   
## [6] "/voter-education/debate-transcripts/october-9-2016-debate-transcript/"   
## [7] "/voter-education/debate-transcripts/october-19-2016-debate-transcript/"   
## [8] "/voter-education/debate-transcripts/october-3-2012-debate-transcript/"   
## [9] "/voter-education/debate-transcripts/october-11-2012-the-biden-romney-vice-presidential-debate/"   
## [10] "/voter-education/debate-transcripts/october-16-2012-the-second-obama-romney-presidential-debate/"   
## [11] "/voter-education/debate-transcripts/october-22-2012-the-third-obama-romney-presidential-debate/"   
## [12] "/voter-education/debate-transcripts/2008-debate-transcript/"   
## [13] "/voter-education/debate-transcripts/2008-debate-transcript-2/"   
## [14] "/voter-education/debate-transcripts/october-7-2008-debate-transcrip/"   
## [15] "/voter-education/debate-transcripts/october-15-2008-debate-transcript/"   
## [16] "/voter-education/debate-transcripts/october-13-2004-debate-transcript/"   
## [17] "/voter-education/debate-transcripts/october-8-2004-debate-transcript/"   
## [18] "/voter-education/debate-transcripts/october-5-2004-transcript/"   
## [19] "/voter-education/debate-transcripts/september-30-2004-debate-transcript/"   
## [20] "/voter-education/debate-transcripts/october-3-2000-transcript/"   
## [21] "/voter-education/debate-transcripts/october-5-2000-debate-transcript/"   
## [22] "/voter-education/debate-transcripts/october-11-2000-debate-transcript/"   
## [23] "/voter-education/debate-transcripts/october-17-2000-debate-transcript/"   
## [24] "/voter-education/debate-transcripts/2000-debate-transcripts-translations/"   
## [25] "/voter-education/debate-transcripts/october-6-1996-debate-transcript/"   
## [26] "/voter-education/debate-transcripts/october-9-1996-debate-transcript/"   
## [27] "/voter-education/debate-transcripts/october-16-1996-debate-transcript/"   
## [28] NA   
## [29] "/voter-education/debate-transcripts/october-11-1992-first-half-debate-transcript/"   
## [30] "/voter-education/debate-transcripts/october-13-1992-debate-transcript/"   
## [31] NA   
## [32] "/voter-education/debate-transcripts/october-15-1992-first-half-debate-transcript/"   
## [33] "/voter-education/debate-transcripts/october-19-1992-debate-transcript/"   
## [34] "/voter-education/debate-transcripts/september-25-1988-debate-transcript/"   
## [35] "/voter-education/debate-transcripts/october-5-1988-debate-transcripts/"   
## [36] "/voter-education/debate-transcripts/october-13-1988-debate-transcript/"   
## [37] "/voter-education/debate-transcripts/october-7-1984-debate-transcript/"   
## [38] "/voter-education/debate-transcripts/october-11-1984-debate-transcript/"   
## [39] "/voter-education/debate-transcripts/october-21-1984-debate-transcript/"   
## [40] "/voter-education/debate-transcripts/september-21-1980-debate-transcript/"   
## [41] "/voter-education/debate-transcripts/october-28-1980-debate-transcript/"   
## [42] "/voter-education/debate-transcripts/september-23-1976-debate-transcript/"   
## [43] "/voter-education/debate-transcripts/september-26-1960-debate-transcript/"   
## [44] "/voter-education/debate-transcripts/october-7-1960-debate-transcript/"   
## [45] "/voter-education/debate-transcripts/october-13-1960-debate-transcript/"   
## [46] "/voter-education/debate-transcripts/october-21-1960-debate-transcript/"

It works. Every element in this vector is a link to a relevant debate transcript. There are two NA’s, which (I believe) is from two “p” elements, that does not contain any “a” elements. This is easily dealt with and is, therefore, not a problem.

Now I just need to automate the process. This is done with a loop which will create an object for each transcript. The loop will not go through NA values. I start by creating an object called “debate\_names”, this will contain the name of each debate object. Then I loop through each element in my previous vector “vect\_link”. I then create an object called “name”. This object contains a substring from the current link. This is used for naming the objects. If the link does not match the regex, it will just contain the full link. Then I create an object called website. It contains the first part of the url and the current link which are concatenated. Then I push the name into my empty vector (debate\_names). I then create an object called “debate”. This contains the scraped data. The debate object is then transformed to a tibble, which is easier to text-mine. Lastly, I create an unique object with the function assign(). This object has the same name as the name in the vector “debate\_names”, and it contains the value of the object “debate”.

debate\_names = NULL #Create an empty object  
  
for (link in vect\_link[!is.na(vect\_link)]) { #For every link in the vector "vect\_link" (Na excluded")  
   
 name <- str\_extract(link, "[A-Za-z]+-\\d+-\\d+") #Create an object called name, set the value to be the regex match.  
   
 if(is.na(name)) { #If name is NA  
 name <- str\_extract(link, ".+") #Set name's value to be the same as the current link  
 }  
  
 debate\_names = append(debate\_names, paste("debate\_",name, sep = "")) #Add the object "name" to the vector "debate\_names"  
   
 website <- paste0("https://www.debates.org/", link) #Create an object called "website".  
 debate <- scrape\_debates(website) #Scrape data from the website and save it in an object called "debate"  
 debate <- tibble(line = 1:length(debate), text = debate) #Make "debate" a tibble, so it can be text-mined  
 assign(paste("debate\_", name, sep = ""), debate) #Create an object with the same name as the element in "debate\_names" which contains the value of debate.  
   
  
}

By now I should have the name for each object in the vector “debate\_names”, let’s check the vector out.

debate\_names

## [1] "debate\_september-29-2020"   
## [2] "debate\_/voter-education/debate-transcripts/vice-presidential-debate-at-the-university-of-utah-in-salt-lake-city-utah/"  
## [3] "debate\_october-22-2020"   
## [4] "debate\_september-26-2016"   
## [5] "debate\_october-4-2016"   
## [6] "debate\_october-9-2016"   
## [7] "debate\_october-19-2016"   
## [8] "debate\_october-3-2012"   
## [9] "debate\_october-11-2012"   
## [10] "debate\_october-16-2012"   
## [11] "debate\_october-22-2012"   
## [12] "debate\_/voter-education/debate-transcripts/2008-debate-transcript/"   
## [13] "debate\_/voter-education/debate-transcripts/2008-debate-transcript-2/"   
## [14] "debate\_october-7-2008"   
## [15] "debate\_october-15-2008"   
## [16] "debate\_october-13-2004"   
## [17] "debate\_october-8-2004"   
## [18] "debate\_october-5-2004"   
## [19] "debate\_september-30-2004"   
## [20] "debate\_october-3-2000"   
## [21] "debate\_october-5-2000"   
## [22] "debate\_october-11-2000"   
## [23] "debate\_october-17-2000"   
## [24] "debate\_/voter-education/debate-transcripts/2000-debate-transcripts-translations/"   
## [25] "debate\_october-6-1996"   
## [26] "debate\_october-9-1996"   
## [27] "debate\_october-16-1996"   
## [28] "debate\_october-11-1992"   
## [29] "debate\_october-13-1992"   
## [30] "debate\_october-15-1992"   
## [31] "debate\_october-19-1992"   
## [32] "debate\_september-25-1988"   
## [33] "debate\_october-5-1988"   
## [34] "debate\_october-13-1988"   
## [35] "debate\_october-7-1984"   
## [36] "debate\_october-11-1984"   
## [37] "debate\_october-21-1984"   
## [38] "debate\_september-21-1980"   
## [39] "debate\_october-28-1980"   
## [40] "debate\_september-23-1976"   
## [41] "debate\_september-26-1960"   
## [42] "debate\_october-7-1960"   
## [43] "debate\_october-13-1960"   
## [44] "debate\_october-21-1960"

It appears to be working. Now I should be able to access each debate by just typing the name of one of the objects above.

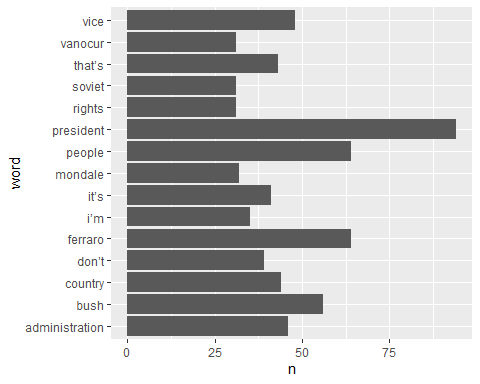
`debate\_october-11-1992`

## # A tibble: 105 x 2  
## line text   
## <int> <chr>   
## 1 1 "\nOctober 11, 1992\n"   
## 2 2 "The First Clinton-Bush-Perot Presidential Debate(First Half of Debate~  
## 3 3 "This is a transcript of the first half of the first presidential deba~  
## 4 4 "Â "   
## 5 5 "LEHRER: Good evening, and welcome to the first of 3 debates among the~  
## 6 6 "PEROT: I think the principal that separates me is that 5 and a half m~  
## 7 7 "LEHRER: Governor Clinton, a one minute response."   
## 8 8 "CLINTON: The most important distinction in this campaign is that I re~  
## 9 9 "LEHRER: President Bush, one minute response, sir."   
## 10 10 "PRESIDENT BUSH: Well, I think one thing that distinguishes is experie~  
## # ... with 95 more rows

It works! Now each debate is scraped, saved in an object and is in a dataframe. Let’s do a very simple text-analysis on one of the debates. First, I need to get a list of stop words, which is included in the tidytext package. Then I tokenize the debate by word so I can do an analysis. I do this with the function unnest\_tokens. I then remove all stopwords from the debate with anti\_join(). Then I count each word and sort it. Then I filter out all words that does not appear more than 30 times. Lastly, I pipe it directly in to a ggplot:

data("stop\_words")#Get all stopwords  
  
 `debate\_october-11-1984` %>% #Chose the debate of october 11th. 1984.  
 unnest\_tokens(word, text) %>% #Tokenize by word  
 anti\_join(stop\_words) %>% #Remove stop words  
 count(word, sort = TRUE) %>% #Count each word and sort the result.  
 filter(n > 30) %>% #Remove words, that does not appear more than 30 times  
 ggplot(aes(word, n)) + #Make a ggplot  
 geom\_col() + #Make it a bar chart where the height represents the value  
 coord\_flip() #Flip the bar chart

## Joining, by = "word"



As you can see, there is still a lot of cleaning to be done, but it is a great start.